

CHINA
OpenStack Days

CHINA
OpenStack Days

IT大咖说
知识分享平台

OpenStack计费框架与落地实践

罗勇
携程云平台高级研发经理



If You Cannot **Measure** it
You Cannot **Manage** it

内容

- 云平台计费需求
- 计费模型与框架选型
- 携程容器计费实现
- Q&A

携程云平台概况

硅谷

伦敦 法兰克福

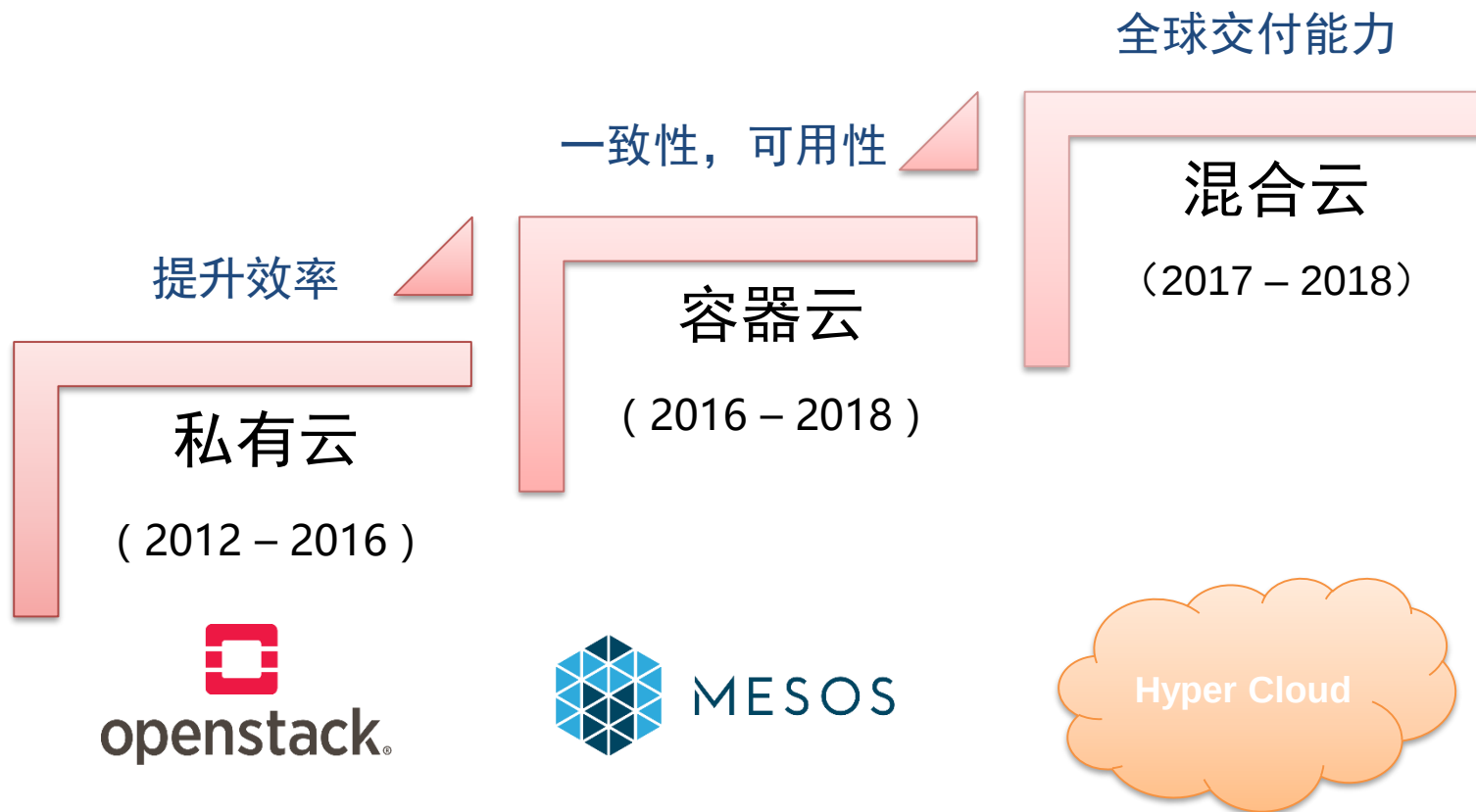
南通 首尔 东京
上海 台湾
香港

曼谷

新加坡

规模：2 区域，4 IDC，9可用区，4K+宿主机
环境：生产，测试
物理机：30+ 机型
虚拟机：VMware、KVM
容器：胖、瘦容器
混合云：9 国家或地区
用户：16 业务部门及兄弟公司

携程云平台演进



过去计费方式

- 按核计量
- 按月付费



问题1：成本分摊是否公平

- 不同负载，小BU低负载，低流量
- 资源消耗成分不同
- 不同环境一样价格
- 月初申请，月底销毁，如何计量
- 容器频繁创建，销毁没法记录
- 混合云分摊

问题2：如何提升资源利用率

- 用户：只管申请，从不释放
- 运维：确定超分比难

部署趋势(生产环境)



问题3： 资源管理困局

- 无容量规划参考依据
- 难做扩容、缩容决策

先聊聊计费那些事儿

计费方式

- 按月、年计费
- 按量计费

对比项	包年包月	按量付费
付款方式	预付费。计费单位为 元/月。	后付费。计费单位为 元/小时。不足 1 小时，按 1 小时计算。
价格	平均每小时价格低。	平均每小时价格高。
带宽	支持固定带宽和按实际流量两种计费方式。	支持固定带宽和按实际流量两种计费方式。
实例规格升级	随时升级。一旦升级，将适用于整个购买周期。请参见 升配 。	不支持升级
带宽规格升级	固定带宽随时升级，至少升级 1 天。	不支持升级
使用场景	适用于固定的 7x24 服务，如 Web 服务。	适用于有爆发业务量，例如临时扩展、临时测试、科学计算等
能否释放	不能提前释放。	可随时释放。请参见 释放实例 。
域名备案	支持	不支持

来自阿里云：https://help.aliyun.com/document_detail/25370.html?spm=5176.product25365.6.576.dRhLaW

记账模型

- 公有云
 - 按量**付费**（人民币）
 - 关注**盈利**
- 私有云
 - 按量**分摊**（比例）
 - 关注**合理性**



核心业务问题

数据精度

秒级、分钟级计量，小时级账单

计费策略

按服务类型，环境，数据中心，折扣率

费用记账归属

按租户，子账号，组织，产品线，应用

实现的三个步骤

Metering
计量

Rating
计费

Billing
记账

OpenStack 计费框架

计费有关项目与现状

- Metric

- Ceilometer
- Gnocchi
- Panko
- Aodn

- Rating

- CloudKitty



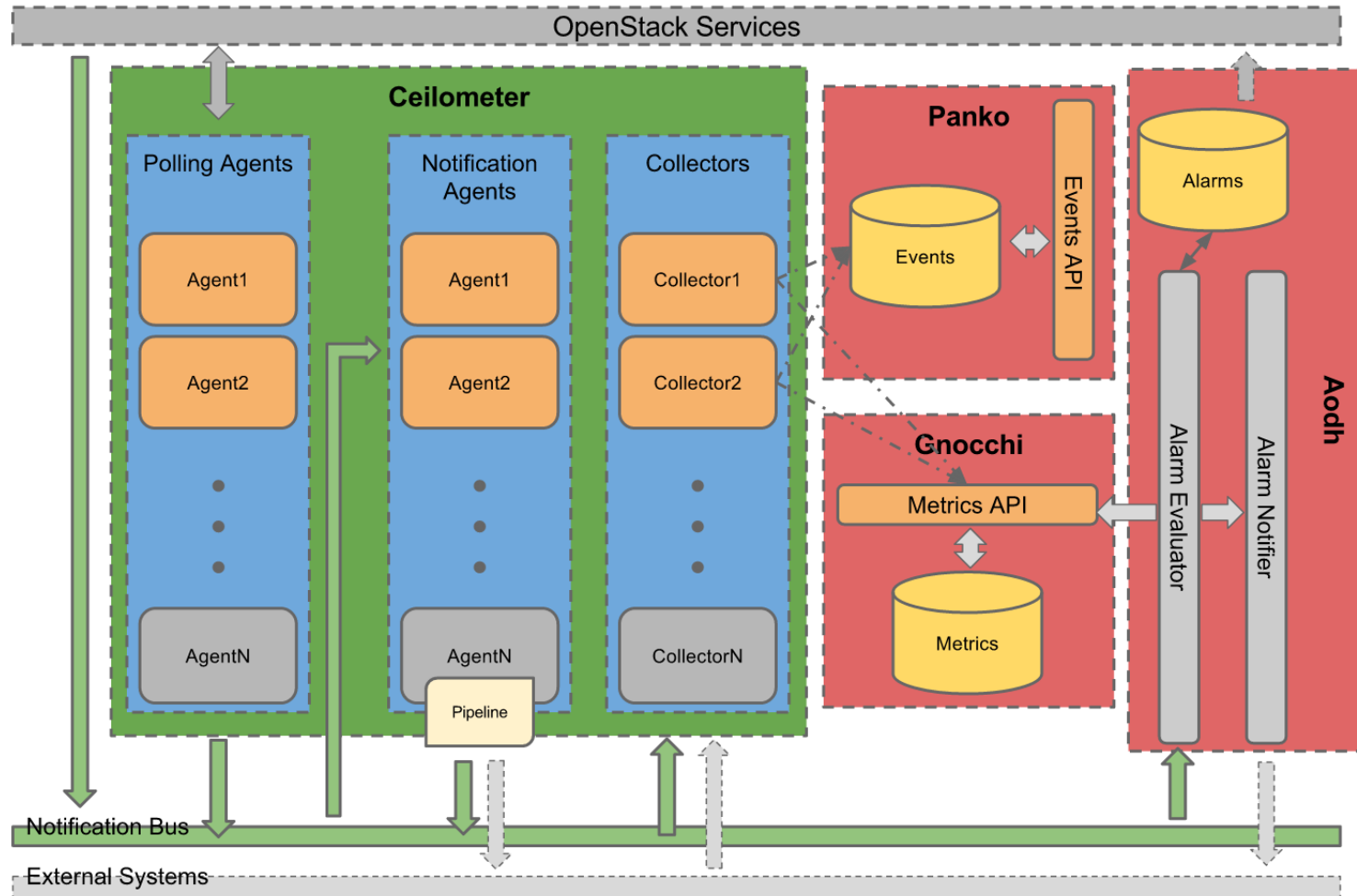
- Billing

- Horizon Plugin

OpenStack Metric

- Ceilometer
 - 采集计量数据并加工预处理
- Gnocchi
 - 存储时序计量数据和提供资源索引
- Panko
 - 提供事件存储服务
- Aodh
 - 提供预警和计量通知服务

OpenStack Metric



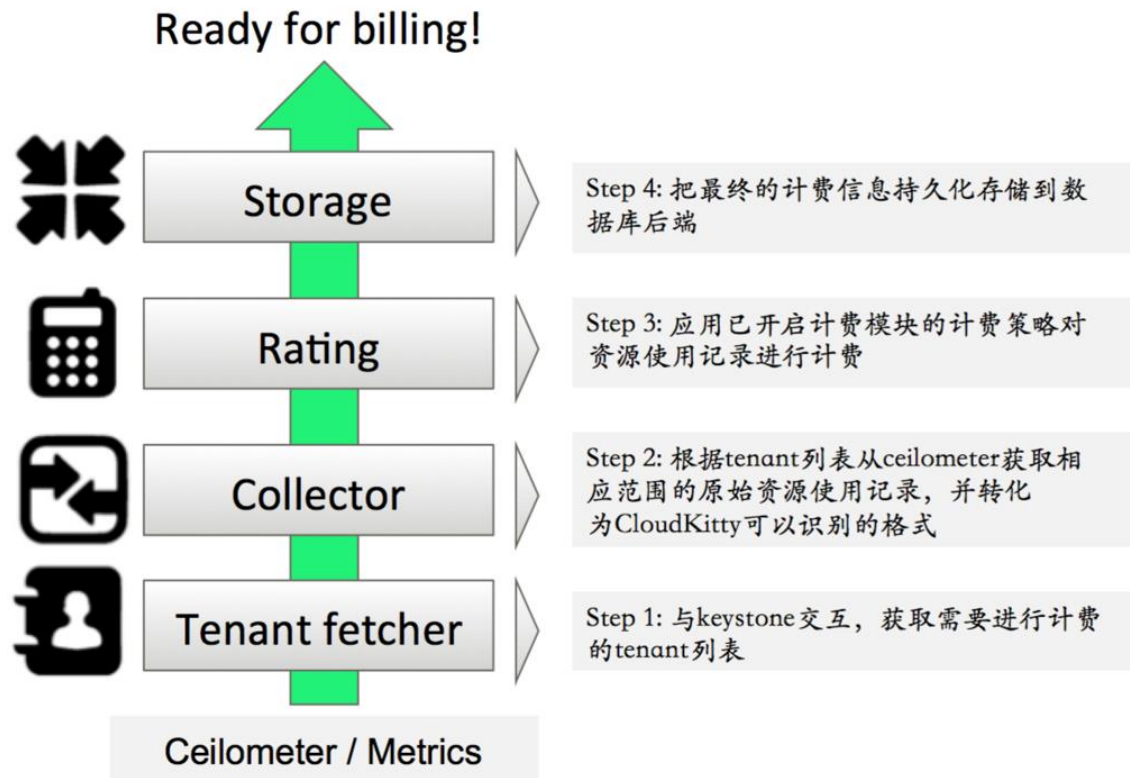
OpenStack Rating

- Cloudkitty 核心模块
 - Tenant fetcher, Collector, Rating, Storage
- Cloudkitty API
 - REST计费接口
- python-cloudkittyclient
 - 命令行交互接口



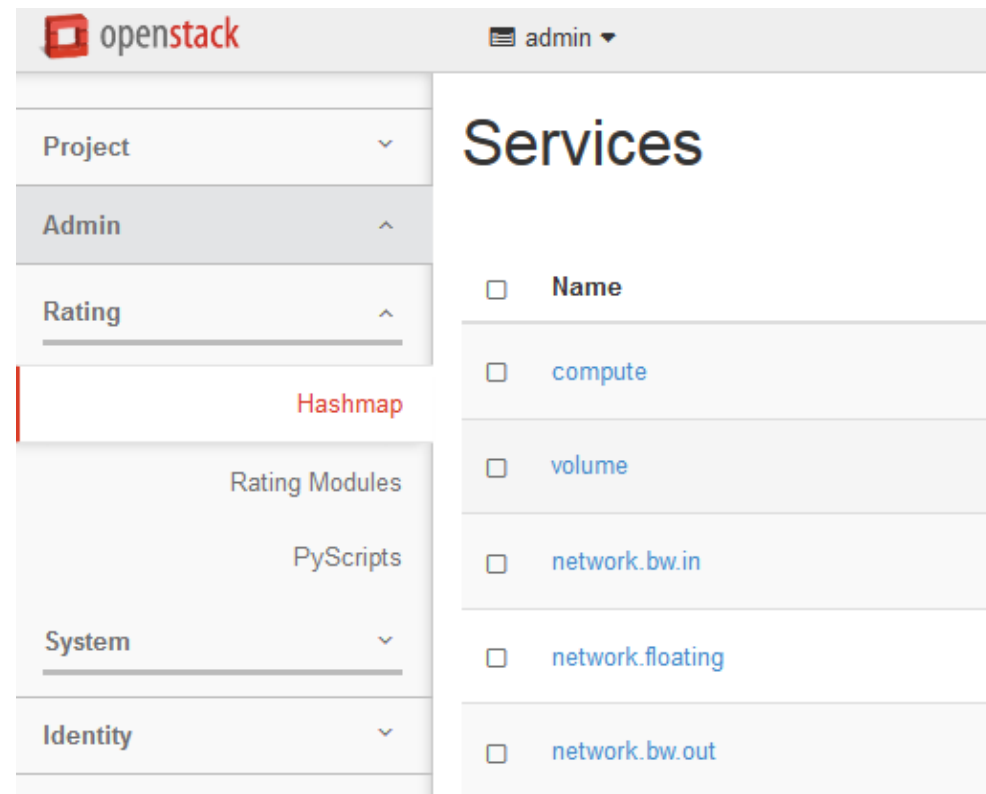
OpenStack Rating

- Cloudkitty Rating 过程



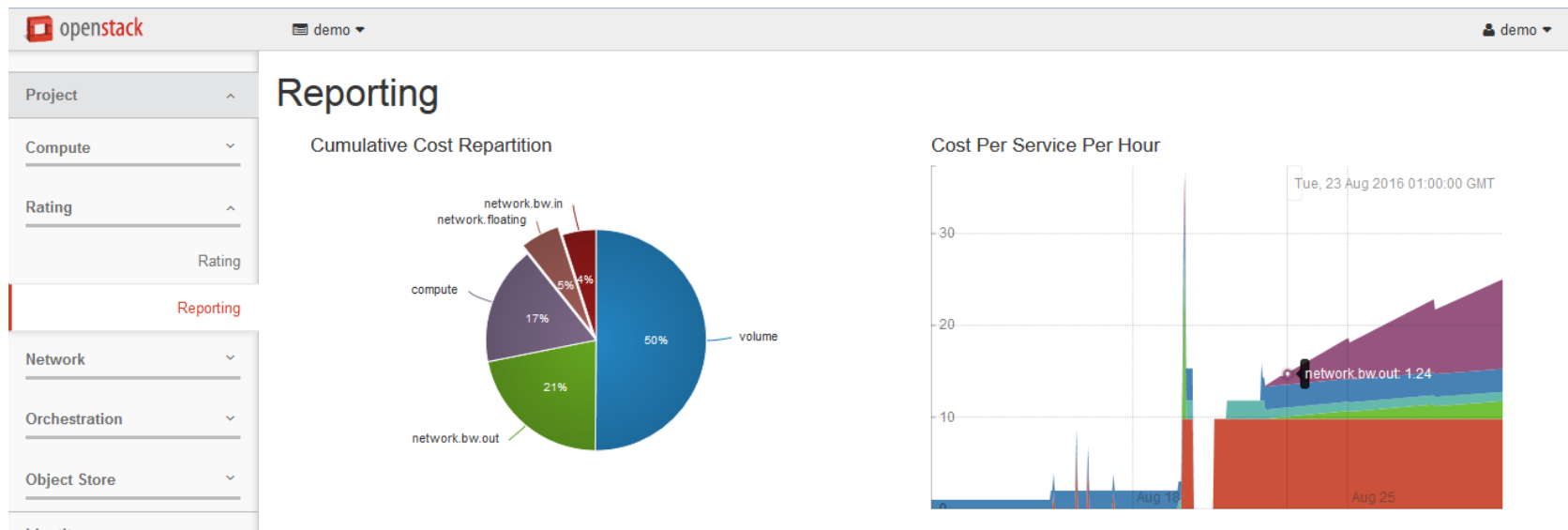
OpenStack Rating

- Cloudkitty 计费模型
 - Noop模型
 - Hashmap模型
 - PyScripts



OpenStack Billing

- CloudKitty-dashboard
 - 提供计费模型配置界面
 - 提供计费查询界面



携程计费实现

方案对比

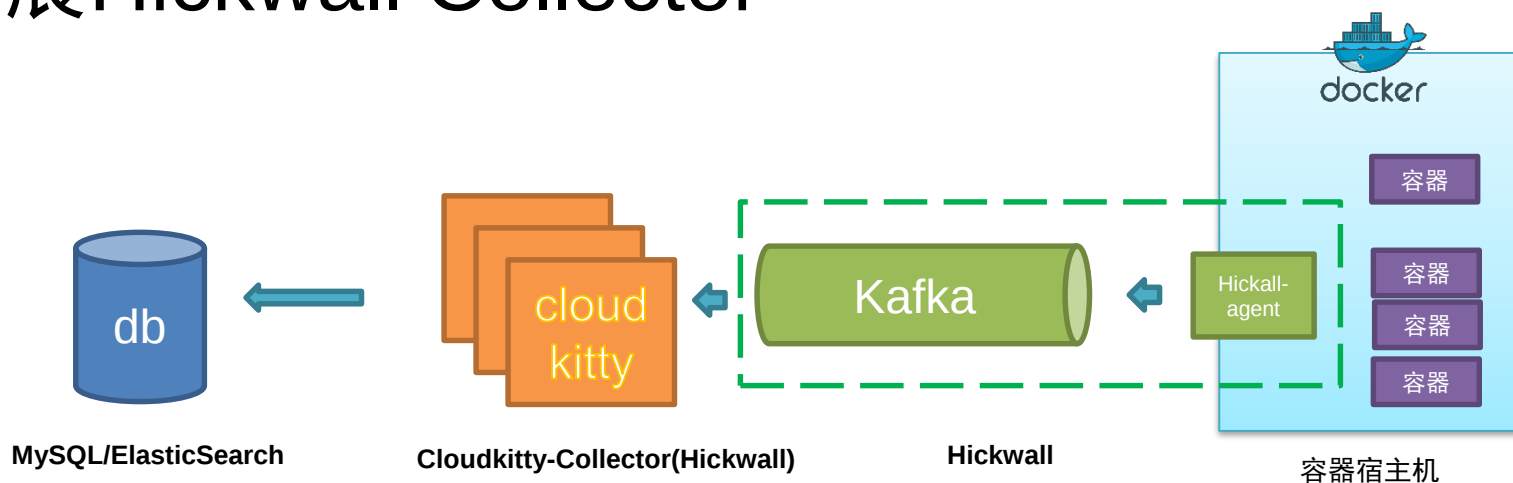
	优点	缺点
A: Ceilometer+Gnocchi+Cloudkitty	1. 针对OpenStack定制, 可以做基于event的计量, 资源从生成到销毁的完整过程都可以详细记录, 可以做到“秒级”计费. 2. 基本上就是社区同步的方案, 可扩展性会比较好, 之后可以扩展来做billing, alarming or capacity planning..	1. Ceilometer 和公司hickwall 功能重叠, 重新发明轮子, 到生产可用的程度周期长 2. 要把新版本的ceilometer改造成I版可用, 估计工作量会比较大 3. 对现有I版OpenStack影响较大, 一次性添加了至少三个组件 4. 宿主机上要增加agent, 收集metric会有额外的cpu负载 5. 对现有OpenStack有负面影响, 提升维护难度.
B: Hickwall + Gnocchi + Cloudkitty	1. 基于Hickwall来获取metric, 对现有系统侵入性小, 风险低. 2. 只需要针对hickwall在cloudkitty中添加一个collector, 代码改动量小. 3. 利用到了gnocchi的优势来解决大数据量导致的性能问题.	1. 依赖于Hickwall收集的数据 2. 计量精度依赖于Hickwall, 最多做到小时级别. 3. 出账单的部分还是需要另外写一个中间件来获取用户和资源的关联关系 4. 目前的设计仅针对billing, 还没有考虑alarming, capacity planning.

计费需求

- Metric 处理 QPS
 - 测试：100宿主机*100实例*8/60秒 = 1.3K QPS
 - 生产：1K宿主机*100实例*8/60秒 = 13K QPS
 - 时延：SLA <1小时
- Rating 数据存储
 - 记录数：13000*3600*24 = 1E9 (1亿条每天)
- 规则灵活性
 - 按主机，环境，数据中心，超分比，应用
- 计费对象扩展
 - 先容器，后虚拟机，物理机，存储，其他

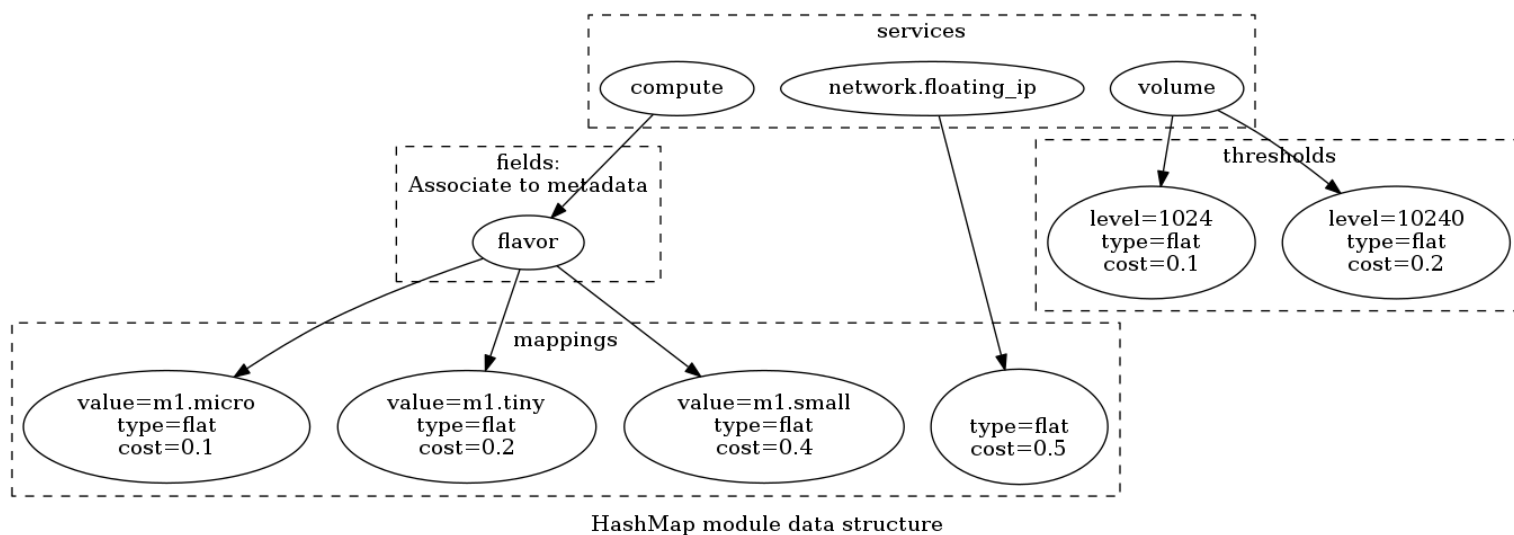
Metric 实现

- 基于携程自研监控平台Hickwall
- 支持秒级、分钟级数据采集
- 支持事件，实时读取数据
- 扩展Hickwall Collector



Rating 实现

- 基于Cloud Kitty 定制
- 启用HashMap 计费模块
- 按服务类型，环境，超分比综合计费



Billing 实现

- 自研Billing Portal
- 按应用，产品，部门记账
- 数据按月，天，时聚合
- 月底对账，平账，控制误差

CloudKitty

- ▶ aggregate day bill for ctrip cloud ▼ aggregate day bill for ctrip cloud ⌚ in 14h55m
- ▶ aggregate day bill for public cloud ▼ aggregate day bill for public cloud ⌚ in 14h45m
- ▶ aggregate hour bill for ctrip cloud ▼ aggregate hour bill for ctrip cloud ⌚ in 35m
- ▶ aggregate month bill for ctrip cloud ▼ aggregate bill_day records into bill_month for all billing(ctrip&other public cloud). ⌚ in 12d15h
- ▶ aggregate year bill for all provider ▼ aggregate year bill for all provider ⌚ in 165d15h

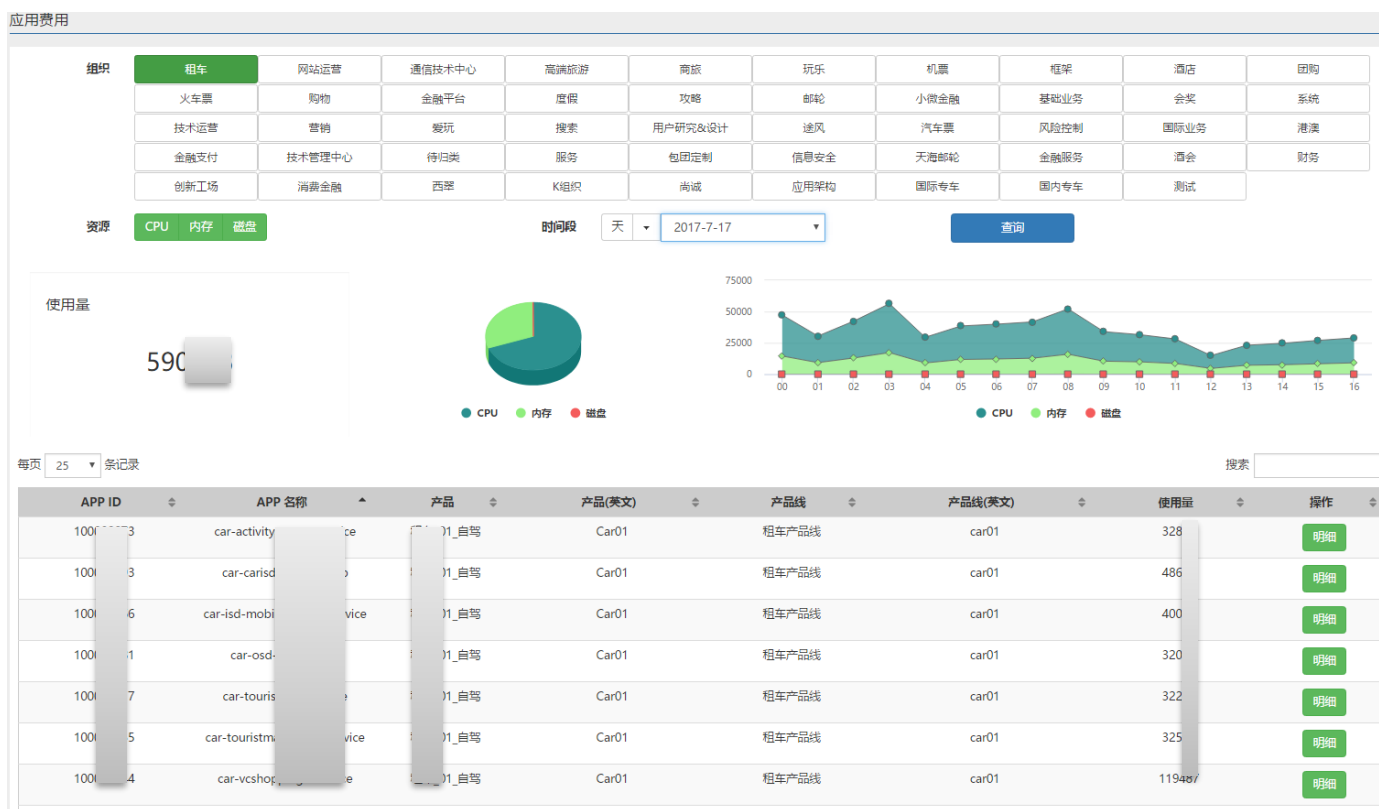
Billing 实现

- 资源分摊比例报表

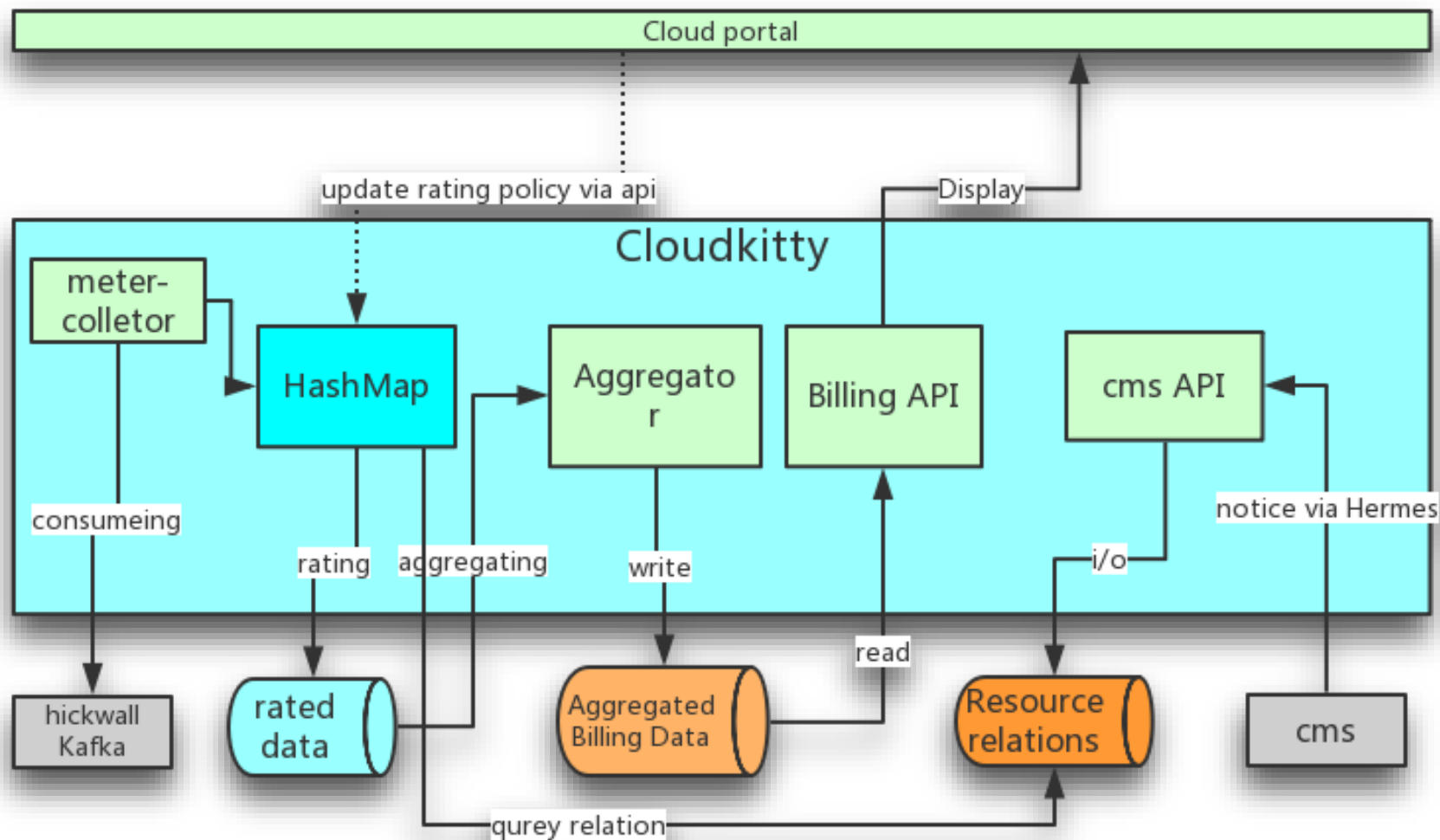


Billing 实现

- 按部门，产品，应用记账



计费系统架构



遇到的一些问题

- Rating吞吐量低，时延过长
 - Pykafka模块balance consumer问题
- 账单查询速度慢
 - 关系数据库 vs 时序数据库
- 对账工作繁琐
 - 海量数据准确性校验，少算好于多算
- 数据准确性监控
 - 基于资源历史告警

未来规划

- Rating Data Store 改造
- 用户自助资源优化
- 覆盖更多计费场景

THANK YOU